~You Only Learn Once ~
A Stochastically Weighted AGGRegation approach to online regret minimization

danny "vato loco" m4turana DIMATURA@CMU.EDU
straight outta south hemisphere

d4vid "the H is silent" f0uhey DFOUHEY@CS.CMU.EDU
212 REPRESENT

Abstract

YOLO YOLO YOLO
2 DEEP 4 U
3 DEEP 5 U

keywords:#yolo, #swag, #swagger, #dat #bound, #machinelearning, #brogrammerproblems, #noregrets, #belieber

Figure 1. #Swagspace.

1. Introduction

We consider the online learning problem, in which the learner receives a life experience x and returns a tumblr post, facebook status, or picture (e.g., self-mirror pic) y. Once the learner makes the post, the internet community responds with a loss function $\ell_t : Dom(Y) \to \mathbb{R}$ that evaluates the swag of the learner's post.

Algorithm 1 you only learn once / online learning

initialize regret $R_t \leftarrow 0$
$t \leftarrow 1$
for $t = 1$ to *death* **do**
  something happens x_t
  post tumblr $y_t \leftarrow f(x_t, y_t)$
  receive loss function $\ell_t(y_t)$
  ~~update regret $R_t \leftarrow R_{t-1} + \ell_t(y_t)$~~ $R_t \leftarrow 0$ yolo lol
  $t \leftarrow t + 1$
end for

Figure 2. #swag #yolo #doublewrapping #doublebagging #STDROCCurve #instagram

In this paper we derive an efficient online learning algorithm, SWAGGR, with pretty tight regret bounds of $O(LOL)$ for every problem. We achieve this by projecting the well-known Randomized Weighted Majority algorithm (Littlestone & Warmuth, 1992) into #swagspace. Swagspace is related to the space induced by the well-known Kardashian Kernel (Fouhey & Maturana, 2012), except in that is accessible to anybody with a smartphone.

Algorithm 2 brogrammers be crushin this code

Input: data x_i, size m
repeat
  Initialize $noChange = true$.
  for $i = 1$ **to** $m - 1$ **do**
    if $x_i > x_{i+1}$ **then**
      Swerve x_i and x_{i+1}
      $noChange = false$
    end if
  end for
until $noChange$ is $true$

Table 1. Equivalences of online learning approaches #socialmedia #instagram

	Facebook	Twitter	Tumblr
Compute Subgradient	Like	Retweet	Reblog
Convex Projection	Comment	Reply	Note

2. On YOLO Learning

We present our YOLO learning framework, SWAGGR, in Algorithm 1. For the sake of notational convenience, we assume that the online community is Tumblr. We have also however have had success using Facebook as well. We show the correspondences between the various problem settings in Table 1.

2.1. Theoretical Analysis - YOLO Regret Bound

We now prove a regret bound on SWAGGR, and demonstrate that for all times $t \in \mathbb{R}^+$ considered by the agent, SWAGGR achieves provably minimal regret. We begin with a review of regret minimization; following this, we present an intuitive and powerful proof of our regret bound. Let $\mathcal{X}$ be an instance space and $Dom(Y)$ be an output space. Let $l_1, \ldots, l_N$ be a set of loss functions with $l_i \in Dom(Y) \to \mathbb{R}$ and let Π be the set of experts from which the algorithm picks a prediction. Finally, let π^* be the expert that in retrospect, incurs the least loss $\sum_{i=1}^{N} l_i(\pi^*)$ over time. We aim to produce a sequence of experts $\pi_1, \ldots, \pi_N$ that minimize the average regret, or difference between our loss and the best expert's loss, or:

$$R_N = \frac{1}{N} \sum_{i=1}^{N} l_i(\pi_i) - l_i(\pi^*) \tag{1}$$

Much research work has been devoted developing algorithms that are no-regret (i.e., $\lim_{N\to\infty} R_T = 0$), and a great deal of effort has been spent on proving this fact for new algorithms. Our YOLO learning bound achieves this no-regret property, and in contrast to past work, the regret bound is easily proved.

Proof. Consider the third-to-last line in Algorithm 1. By definition the instantaneous regret is zero, and so the average regret is also zero. #YOLO #2SWAG4U □

Note that previous work has either focused on convex sets of experts or has showed results that only hold for small numbers of experts. By adhering to the SWAG philosophy and ignoring regret, we achieve state-of-the-art performance without such limitations.

3. Application - SVM Learning with SWAGGR

It is well known that in Swagspace one does not need condoms (see Fig. 2); we present an analogous analysis for Support Vector Machines (SVMs). One success story of online convex programming is the development of online solutions to the SVM problem, obviating the use of complex and expensive quadratic programming (QP) toolkits. A generalization of our SWAGGR algorithm also allows the general solution to all quadratic programming problems. We compare a high-SWAG approaches drinking 4-LOKO (a high alcohol and caffeine beverage) with a state-of-the-art quadratic programming solver, LOQO (Vanderbei, 1999) in Table 2. LOQO is better in only one category (solving QPs), and 4-LOKO is better in 5 (alcohol content, swag, etc.). Clearly, the solution is to use 4-LOKO. Anecdotal evidence confirms that 4-LOKO is indeed good at getting people to local minima (e.g., falling into ditches, etc.). #crunk

4. Discussion and Future work

The YOLOSWAGGR algorithm may in fact lead to high regret later on in life. But that is beyond the current planning horizon of most teenage agents. It seems that perhaps it is necessary to accumulate regrets; then, when brain development reaches adulthood, these regrets can be processed to form better policies. This superficially resembles the practice of accumulating subgradients and taking a step in the average direction, suggesting the validity of our approach.

References

Fouhey, David F. and Maturana, Daniel. The Kardashian Kernel. In SIGBOVIK, 2012.

Table 2. How to choose a Quadratic Programming Toolkit. We present a comparison of LOQO (Vanderbei, 1999) and our approach, 4-LOKO. Clearly 4-LOKO is better for solving QPs. #crunk #sizzurp #geTtiNItIn

	Solves QPs	Refreshing	SWAG	Banned by n states	% Alcohol	# LOKOs
LOQO	✓	X	0	$n = 4$	0%	1
4-LOKO	X	✓	5	$n = 50$	12%	4

Figure 3. #biebs #teen #swag #cute

Littlestone, Nick and Warmuth, Manfred K. The weighted majority algorithm. In IEEE Symposium on Foundations of Computer Science, 1992.

Vanderbei, Robert. Loqo user's manual – version 3.10. Optimization Methods and Software, 12:485–514, 1999.

Figure 4. #ferret #swag #yolo #class

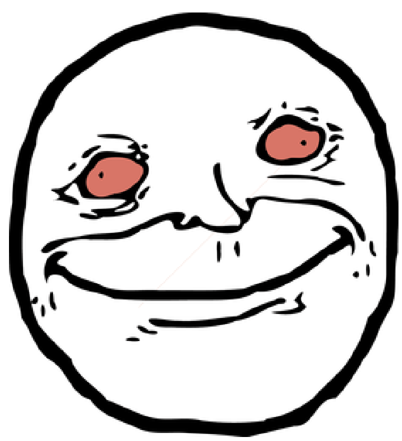

Figure 5. Le Me, A maChiNE leARNer wit Mad boOsted deCISion TREES n smokin like a a max-ent pRIoR. WhEre u 3quentists noW? #iceburn #swag #classy #enlighted-bymyownintelligence #euphoric